

MEMORY CONTROL METHOD AND APPARATUS

Patent number: JP1306917
Publication date: 1989-12-11
Inventor: MUUSUDASHI SHIYAWANKAA MENON
Applicant: IBM
Classification:
 - International: G06F3/06; G11B20/12
 - European: G06F3/06F; G06F3/06M; G11B20/12M
Application number: JP19890078721 19890331
Priority number(s): US19880197057 19880520

Also published as:

- EP0347032 (A)
- EP0347032 (A)
- EP0347032 (B)

[Report a data error here](#)

Abstract of JP1306917

PURPOSE: To improve the access performance of a record to be emulated by reforming the data record of a first format to a second non-interchangeable recording format while maintaining the byte displacement relative position of the record of the first format in the second format. **CONSTITUTION:** At the time of correcting so as to be stored by the second format, data of the first format including a control field and a data field is corrected to be stored in each address possible record as data of the second non-interchangeable mode while maintaining the byte displacement address possibility of the record of the first format.

Correction like this attains the emulation of the first format to the second format by maintaining a gap between records by both formats while removing a gap within the record, error detecting redundant information, physical parameter data and a padding characteristic to the first format during correcting. Thereby, the access performance of the record to be emulated is improved.

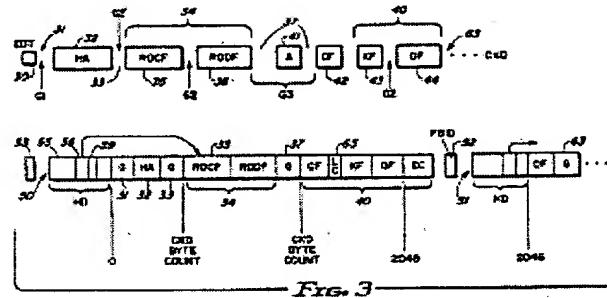


Fig. 3

Data supplied from the **esp@cenet** database - Worldwide

⑫ 公開特許公報 (A)

平1-306917

⑬ Int. Cl.

G 06 F 3/06
G 11 B 20/12

識別記号

301

府内整理番号

Z-6711-5B
8524-5D

⑭ 公開 平成1年(1989)12月11日

審査請求 有 請求項の数 3 (全16頁)

⑮ 発明の名称 記憶制御方法及び装置

⑯ 特 願 平1-78721

⑰ 出 願 平1(1989)3月31日

優先権主張

⑱ 1988年5月20日 ⑲ 米国(U S) ⑳ 197057

㉑ 発明者	ムースダシ・シャワン カー・メノン	アメリカ合衆国カリフォルニア州サン・ホセ、モントロ・ プレス6017番地
㉒ 出願人	インターナショナル・ ビジネス・マシーン ズ・コーポレーション	アメリカ合衆国10504、ニューヨーク州 アーモンク (番 地なし)
㉓ 代理人	弁理士 山本 仁朗	外1名

明細書

1. 発明の名称 記憶制御方法及び装置

2. 特許請求の範囲

(1) 複数のレコードからなる第1の様式のデータを、該第1の様式のまま、第2の様式の記憶手段に記憶するための、下記ステップ(イ)ないし(ニ)を含む記憶制御方法、ただし、上記第1の様式のデータは、それぞれアドレス可能な第1の制御フィールド及び第1のデータ・フィールドとフィールド間ギャップとを各レコード内に含むと共に、相次ぐレコードの間にレコード間ギャップを含み、各レコードのフィールド及びギャップのアドレス可能な位置は第1の様式のデータの始点を基準とするバイト・カウントで識別可能であり、かつ各レコードの制御フィールドは特定の制御に関係する部分を含んでおり、さらに、上記第2の様式の記憶手段は、それぞれ所定のバイト数の記憶容量を有する複数のアドレス可能なブロックを含むものとする：

(イ) それぞれ上記アドレス可能なブロックに等しい記憶容量を有する複数のアドレス可能な領域をパッファ手段に設定し、

(ロ) 上記第1の様式のデータから、上記特定の制御に関係する部分及びフィールド間ギャップを除去し、

(ハ) 上記第1の様式の残りのデータ及びレコード間ギャップを上記パッファ手段の複数のアドレス可能な領域に記憶し、

(ニ) 上記記憶手段のアドレス可能な各ブロックが上記パッファ手段の対応する領域からのデータを受け取り、かつ上記第2の様式での始点から各レコードの先頭までのバイト変位が上記第1の様式のデータの始点を基準とするカウント・バイトに等しくなるように、上記パッファ手段内のデータ及びレコード間ギャップを上記記憶手段に記憶する。

(2) 制御フィールド、データ・フィールド及びフィールド間ギャップをそれぞれ有する複数のレコードとレコード間ギャップとを含みかつ上記

フィールド内に誤り訂正、充填、及び物理的パラメータのための特定部分を含むデータを供給するデータ供給手段と、所定数のバイトをそれぞれ記憶しうる複数のアドレス可能な固定ブロックを有する記憶手段とを有する情報処理システムにおいて、上記記憶手段におけるデータの記憶を制御するための、下記ステップ(a)ないし(e)を含む記憶制御方法：

(a) 上記データ供給手段によって供給されるデータから上記特定部分及びフィールド間ギャップを除去し、

(b) 上記データの最後のレコードの制御フィールドに最終レコード標識を付加し、

(c) 上記データにおける順序と同じ順序でかつ上記レコード間ギャップを維持しつつ上記データの複数のレコードを配列し、その際、配列後のデータの始点から各レコードの先頭までのバイト単位の変位が元のデータにおける始点から各レコードまでのバイト単位の変位に等しくなるようにし、

(d) 配列したデータを、元のデータの様式に

- 3 -

レータ手段が、

上記所定の記憶容量に等しい記憶容量をそれぞれ有する複数の領域を有するバッファ手段に上記カウント・キー・データ様式のデータを一時的に記憶し、そこから上記記憶装置へ転送する手段と、

仮想トラックの始点からカウント・キー・データ様式の各レコードの先頭までのバイト変位を、カウント・キー・データ様式のトラックの始点から各レコードの先頭までのバイト変位と等しくするようにして、上記データを上記記憶装置の複数の固定ブロックに記憶するための仮想トラックを設定する手段と、

上記仮想トラックの外において、各固定ブロックの先頭にヘッダを設け、該固定ブロックにカウント・キー・データ様式の第1のレコードが含まれるときには、該第1のレコードまでの変位を示すポインタを上記ヘッダに記録し、該固定ブロックにカウント・キー・データ様式の第1のレコードが含まれないときには、そのことを示す情報を上記ヘッダに記録する手段と、

- 5 -

かかわりなく、上記記憶手段の固定ブロックに対応する長さに分割し、

(e) 分割したデータを上記記憶手段に書き込む。

(3) それぞれ所定の記憶容量を有する複数の固定ブロックを有する固定ブロック様式の直接アクセス型記憶装置と、カウント・キー・データ様式に関連したプログラムを用いる上位プロセッサと、該上位プロセッサと上記記憶装置との間に設けられたチャネル・プロセッサとを有しあつ該チャネル・プロセッサが上記上位プロセッサと上記記憶装置との間のデータ転送及びそれに関連した動作を命じる複数のチャネル・コマンドを生じる制御部を有する情報処理システムにおいて、上記記憶装置を制御する装置であって、

上記チャネル・プロセッサの制御部と上記記憶装置との間に設けられていて、所定のチャネル・コマンドに応答して、上記記憶装置においてカウント・キー・データ様式のデータの記憶処理をエミュレートするエミュレータ手段を有し、該エミュ

- 4 -

を有する記憶制御装置。

3. 発明の詳細な説明

A. 産業上の利用分野

本発明は、情報処理システムの記録動作に関し、具体的には、第1の様式とは互換性のない第2の様式でフォーマットされた記録装置に第1の様式で配列されたデータを記録することに関する。.

B. 従来技術及び発明が解決しようとする問題点

情報処理システムは、周辺データ記録装置にデータを記録するために多様な様式を使用してきた。あるデータ記録様式が選択されると、情報処理システムのプログラミングが、選択された様式で動作するように設計される。この制限は、データ記録装置が、プログラミングで企図された様式を使用しなければならず、またその逆もあり得ることを意味している。ある記録様式を他の記録様式でエミュレートすることが実現されているが、こうしたエミュレーションでは、プログラミングがエミュレートされた様式を容易にアドレスできなかった。すなわち、エミュレーション・ソフトウェア

- 6 -

は、第1の様式のアドレスを第2の様式の異なるアドレスに変換しなければならなかった。こうしたアドレス変換は、第2の様式に記録されたとき第1の様式のレコード・アドレスを混乱させるだけでなく、かなりのバッファリング資源と命令実行資源を必要とする。

マリリン・ポール (Marilyn Bohl) 著「IBM 直接アクセス記憶装置入門 (INTRODUCTION TO IBM DIRECT ACCESS STORAGE DEVICES)」、出版物 S R 2 0 - 4 7 3 8 、 Science Research Associates, Inc. 、シカゴ、の第4章には、2つのデータ様式が記載されている。第1の様式は、インターナショナル・ビジネス・マシーンズ社 (IBM) のディスク記録装置 (ディスク・ファイル及びDASDとも呼ばれる) で使用される周知のカウント、キー、データ (CKD) 様式である。多くの大規模情報処理システムが、そのプログラミング及び記録にCKDデータ様式を利用している。ポールはまた、固定ブロック (FB) 記録 (ブロック・アーキテクチャ (FBA) 様式と

- 7 -

以前のFBA装置でCKD様式データを記録する手順は、「IBM 4321/4331 プロセッサ互換性機能 (IBM 4321/4331 Processors Compatibility Features)」、IBM 出版物 G A 3 3 - 1 5 2 8 - 2 、第3版、1982年9月、に一般的に記載されている。この様式重複記録では (真のエミュレーションではない) 、IBM 2 3 1 1 及び 2 3 1 4 DASD ボリューム上の CKD 様式データが、IBM 3 3 1 0 FBA 記録装置に記録された。この出版物の教示によると、CKD 様式データをFBA 記録装置に圧縮するため、すべてのギャップ (DASD に記録されたときデータ様式の様々なフィールド間に挿入されたデータを含まない領域) が除去される。この動作は、CKD 様式によるレコードの CKD アドレス可能性を混乱させる。CKD 記録装置のトラック全体 (そのトラックがデータで満たされていない場合でも) が FBA 記録装置に記録される。様式重複記録のマッピングが、前記の出版物の第2-3ページに示されている。FBA 記録装置に記憶された

- 9 -

も呼ばれる) も記載している。対照的に、CKD 様式では、レコード・サイズに応じてディスク・トラック記録様式を分割することにより、大きなレコードのディスク空間を効率よく使用できる。FBA 様式では同一サイズの1組のアドレス可能領域 (ブロックと呼ばれる) を使用し、その領域のそれぞれに、512 バイト、1024 バイト、2048 バイトなどある固定量までのデータが記憶できる。(512 バイトより小さいものなど) 小さいサイズのレコードでは、FBA は CKD に比べていくつかの利点をもつ。いずれにせよ、一部は CKD 様式を使用し一部は FBA 様式を使用する DASD を備えた情報処理システムが利用される。システムで実行されるプログラムの多くは、CKD 様式で動作するように設計されている。したがって、CKD 様式データを FBA 装置で記録できるようにすることが望ましい。FBA 様式で記憶されたすべてのデータ・レコードの CKD レコード・アドレス可能性を維持することも望ましい。

- 8 -

CKD レコードの CKD 回転アドレス可能性を維持することにより、FBA 様式の記録装置で CKD 様式の真のエミュレーションを実現することが望ましい。

本発明の目的は、第2様式において第1様式のレコードのバイト変位相対位置を維持しながら、第1の様式のデータ・レコードを第2の非互換レコード様式にフォーマットし直すことにある。

本発明の他の目的は、第2の様式でのアドレッシングで仮想トラック概念を使用して、両方の様式でレコードのバイト変位アドレスを同じに維持しながら、制御情報を第2の様式に追加することにより第2の非互換様式で記録されたときに第1の様式のデータのアドレッシングを提供することである。

C. 問題点を解決するための手段

本発明によれば、第2の様式で記憶されるように修正されたときに第1の様式のレコードのバイト変位アドレス可能性を維持しながら、各アドレス可能レコード中に (CKD 様式データ中のカウ

- 10 -

ント・フィールドやキー・フィールドなどの)制御フィールドとデータ・フィールドを含む第1の様式のデータが、第2の非互換様式のデータとして記憶されるように修正される。こうした修正は、その間に、レコード内ギャップ、エラー検出冗長情報、物理パラメータ・データ、第1の様式に固有のパディングを除去しながら、両方の様式でレコード間ギャップを維持することにより、第1の様式の第2の様式へのエミュレーションを達成する。第2の様式でのデータ構成は、第1の様式から独立し、レコードが第2の様式のアドレス可能部分に分割されるようになっている。

制御情報は、アドレス可能部分に記憶された1つのレコードの1つの制御フィールドに関係する第2の様式の各アドレス可能部分に追加される。第1の様式のデータ長は完全に可変であり、最小数のアドレス可能部分が修正された第1の様式のデータを第2の様式で記憶する際に消費される。

本発明の特定の実施例では、CKD様式データの1つまたは複数のDASDトラックが、FBA

- 11 -

入出力マシン動作を実行するため、主記憶装置11に記憶されたチャネル・プログラムを実行する。こうしたチャネル・プログラムは、それぞれIBM製造のいわゆる370アーキテクチャのコンピュータで実施されている、チャネル制御ワード(CCW)のリストである。こうした通常のチャネル・プログラムの実行は、チャネル・プロセッサ12のチャネル実行部13で表わされる。チャネル・プロセッサ12は、FBA装置またはDASD16に接続されたFBAアダプタ15に通常のケーブル配線を介して接続されている。FBA様式によるDASD16との間でのデータの記憶及び検索は、周知の周辺データ記憶技術による。DASD16は、複数の同時回転ディスク17を含み、それらのディスクはそれぞれ、周知のように、1つまたは2つの記録面をもち、そこに多数のアドレス可能レコード・トラックがシリンドとして配列されている。

上位プロセッサ10によって実行されるプログラムの多くは、周知のCKDデータ様式を使用す

様式のブロックが第2の様式のアドレス可能部分となる仮想トラックとしてFBA様式でエミュレートされる。この実施例は、データの多くの仮想トラックで構成される仮想CKDディスクをエミュレートすることを含む。

本発明の上記及びその他の目的、特徴及び利点は、添付図面に示す、本発明の好ましい実施例についての以下の具体的な説明から明らかになるであろう。

D. 実施例

図面を参照すると、同じ部品及び構造上の特徴を同じ番号と文字で示す。第1図に、本発明を利用すると好都合な情報処理システムが、主記憶装置11からのプログラムを実行する上位プロセッサ10を含むものとして示されている。周辺プログラム記憶装置(図示せず)からのプログラムを主記憶装置11にページングするには、通常の手順を使用する。(通常は上位プロセッサ10と同じキャビネットに実装されている)チャネル・プロセッサ12が、上位プロセッサ10の望ましい

- 12 -

るので、こうしたCKD様式データはDASD16に直接記録できず、通常CKD様式のDASDが使用される。上位プロセッサ10のプログラムをFBA様式のDASD16に関連して良好に動作させるため、チャネル実行部13とFBAアダプタ15の間にエミュレータ20が挿入され、CKD様式化データをFBA様式でエミュレートする。エミュレータ20は半導体チップの形の論理回路でよく、ROM形式、ディスケット形式、または周辺プログラム記憶装置からチャネル・プロセッサ12へのページングによってプログラミングされる。こうしたエミュレータは、チャネル・プロセッサ12の回路によって実行されるように主記憶装置11に格納してもよく、チャネル・プロセッサ12の個別プログラム・メモリに格納してもよい。

CKDデータは仮想データ・トラックとしてのFBA様式のDASD16に記憶される。CKDデータは1つまたは複数のCKDデータ・トラックにある。仮想トラック中のバイト数は、CKD

- 14 -

装置またはFBA装置の物理トラック・サイズには依存せず、CKD様式データのサイズによって仮想トラックのサイズが決まる。CKDデータ中のバイト数は既知、または容易に計算できる（上記ボール論文参照）。以下の説明では、DASD 16のディスク17がCKD様式データを受け取るようにフォーマットされていると仮定する。前記の「CKDエミュレーションのためにFBAディスクを初期化するステップ」の流れ図を参照。エミュレーションを作成する最初のステップは、データを記憶するのに必要なDASD 16のトラックのブロック数を計算することである。CKDデータのバイト数をXとし、DASD 16の1つのアドレス可能ブロックに記憶できるバイト数をBPBとする。さらに、エミュレートされたトラックまたは仮想トラックのCKDレコードへの直接アクセスを容易にするため、各ブロックに制御情報ヘッダHDを記録することが望ましい。ヘッダ用に使用されるバイト数をCIDとする。次に、FBA様式のB個のブロック（Bは(X/(BPB-

- 15 -

仮想トラック・バッファ26は、主記憶装置11の一部として割り振られる。新しい大型の上位プロセッサでは、チャネル・プロセッサ12はそれ自体のメモリをもつ。その場合は、バッファ26はチャネル・プロセッサ12のメモリの一部として割り振られる。各項目領域27は、FBAブロックの記憶容量に等しい記憶容量をもつ。各項目の一一番左の2つのバイトは、後で説明する制御情報ヘッダHDを記憶し、各バッファ項目の残りのバイトはFBA様式にしたCKDデータを記憶する。HDは、各FBAブロックのバイト位置0と1に記録される。CKDデータは、各FBAブロックのバイト位置2以降に記録される。エミュレータ・プログラム25は、第3図に示すバッファ26の項目に当てはまるようにCKDデータを処理する。エミュレーションは、オリエンテーション・テーブル28を用いて仮想トラックの仮想回転オリエンテーションを行なうエミュレーション・プログラム25によって完了される。この仮想オリエンテーションは、バッファ26に記憶されたデータ

- 17 -

CID）を上の整数に切り上げる。FBA様式のDASD 16のB個のブロックを、CKDデータを記憶するために割り振る。こうした割振りは、通常の処理手順に従う。1つのFBAトラックを使用するのかそれとも複数のトラックを使用するのか、その割振りが断片化され、断片が互いに連結してリストされるかどうか（断片化すると性能が低下する）、その割振りがDASD 16の指標で始まるのか、それともトラック終端（EOT）で始まるかどうかは問題ではない。

CKDデータの修正は、修正の流れを示す第2図とエミュレーションのフォーマット化を示す第3図を参照すると最もよく理解できる。この説明では、エミュレータ20はチャネル・プロセッサ12で実行できる1組のプログラム25として実施される。これらのチャネル・プログラムを、この説明の後ろの部分で流れ図として示す。本発明は、本発明にしたがって構成されたプログラムをFBAアダプタ15など周辺サブシステム制御装置に格納することによっても容易に実施される。

- 16 -

に関するものである。CKDデータがバッファ26内でFBA様式にエミュレートされると、バッファ26の内容がFBAアダプタ15からDASD 16に送られて、割り振られたFBAブロックに記憶される。構成された実施例では、割り振られエミュレータ20でフォーマットされたFBAブロックの内容が、こうしたブロックがCKDデータを含むか否かにかかわらず、まずDASD 16からバッファ26に読み取られる。

第3図は、CKDデータが真のエミュレーションでFBAブロックに記憶されるCKD様式及びFBA様式を示す。CKDデータ様式は、トラック終端（EOT）標識30（CKD DASDでは、EOTは指標である）を含む。仮想トラックでは、EOTがトラックの始めと終わりを示す。すなわち、論理的には仮想トラックは円形である。位置31にあるレコード間ギャップG1が、EOTをCKDトラックで常に見られる第1のレコードHA32から分離する。第2のレコード間ギャップG2、33がHA32をR0レコード34か

- 18 -

ら分離する。RCレコード34は、制御情報R0カウント・フィールド(ROCF)35とR0データ・フィールドR0DF36を含む。第3のレコード間ギャップG3 37が、レコードR0 34を第1のデータ・レコードR1 40から分離する。物理アドレス・マークA41が相次ぐレコードの間ににおいてCKD物理トラックのギャップG3に挿入される。CKD物理アドレス情報Aがエミュレーションから削除される。CKDレコード1は、カウント・フィールドCF42、キー・フィールドKF43及びデータ・フィールドDF44を含む。KF43は任意選択のフィールドである。レコードR1 40中の制御情報は、CF42とKF43に含まれている。CKDデータ中の後続のすべてのレコードR2などは図示した周知の様式を使用する。

エミュレーションのためのCKDデータを記憶するFBA様式トラックが、FBAブロック50及びそれに続くFBAブロック51として示されている。ブロック50は、CKDデータの第1の

- 19 -

ングはCKDトラック上のレコードの回転位置であることに留意されたい。仮想トラックはFBAで物理的に実現されているので、真のエミュレーションには、この同じバイト変位が仮想トラックで発生することが必要である。ブロック50は、その最初の2つのバイト0と1が制御ヘッダHD55で占められている。各HDは、当FBAブロックに記憶される最初に発生したカウント・フィールドの第1バイトを指す、バイト変位ポインタ56を含む。FBAブロック50中で、ポインタ56はROCF、すなわちレコードR034のカウント・フィールド35の第1バイトを指す。あるFBAブロックがCKDカウント・フィールドを記憶していない場合、ポインタはヌルで、そのFBAブロックでレコードが始まらない(カウント・フィールドなし)ことを示す標識ビット59が設定される。この説明の範囲外の他の制御情報もHD55に含まれる。2バイト(FBAブロックの始めから数えて3バイト)目からは、ギャップG1 31が始まる。このレコード間ギャップは、

- 21 -

部分を記憶するものとして示されている。ブロック50は、FBAトラックのどの円周上の位置にあってもよい。FVID(固定ブロック物理識別)52がFBAブロック50と51の間にある。FVID52は後続のブロックのトラック・アドレスとブロック・アドレスを記憶する。FVID53はFBAディスク上のブロック50の位置を物理的に識別する。このFBA物理識別は、DASD上でCKD様式で使用されるCKD物理識別と置き換わる。こうした置換えにより、物理DASD位置を利用する回復処理が可能になる。後で明らかになるように、データの始めからのバイト変位で表わしたCKDレコード・アドレスは、FBA物理アドレスと所定の既知の関係をもち、FBA様式ディスクに記憶されたCKDレコードへの直接かつランダムなアドレッシングが可能になる。

FBAブロック50は、CKDレコードのCKDバイト変位アドレッシングを維持するためのCKD様式データのパッキングの例である。CKD様式ディスク上でのCKDバイト変位アドレッシ

- 20 -

レコードHA32の最初のバイトがFBAブロック50内で、HA32がCKD様式でEOT30からずれているのと同じバイト変位で始まることを示す、正しい数のギャップ・バイト(例えばゼロ)をもつ。CKDレコード間ギャップ33は、FBAブロック50でもHA32とR0レコード34の間で同様に複製されており、ROCF35の第1バイトが、FBAブロック50のバイト2(仮想CKDトラックの始め)から、CKD様式でEOTからずれているのと同じ変位をもつ。後続のすべてのレコード間ギャップ37と63も同様に、CKDトラックの始め及び仮想トラックの始めからの各CKDカウント・フィールドの第1バイトのバイト変位を実現するようなサイズになっている。実際の数値計算は上記のボール論文に記述された2つの様式から容易に明らかになるので、簡潔にするために計算を省略する。

仮想トラックのFBAブロックに対する物理的関係は比較的単純である。各FBAブロックは2048バイトを記憶できるものとして示されてい

- 22 -

る。HD55に2バイトが使用されるので、各FBAブロックは仮想トラックの2046バイトを記憶する。FBAブロック50はバイト0ないし2045を記憶し、次の隣接FBAブロック52はバイト2046ないし4091を記憶し、第3のFBAブロック（図示せず）はバイト4092ないし6137を記憶し、以下同様にして、すべての後続FBAブロック（図示せず）が仮想CKDトラックとしてCKDデータを記憶する。割り振られた1組のB個の（Bは整数）FBAブロック中の最後のFBAブロックは、仮想トラックによっておそらく完全には使用されない。そうした場合、埋込みバイトが最後に割り振られたFBAブロックの未使用部分を充填する。

CKDトラック中にある制御情報の大半は削除され、物理的FBAブロックの仮想CKDトラックには記録されない。各FBAブロックは、通常それ自体の誤り検出訂正（ECC）冗長バイト、それ自体の埋込みバイト、それ自体の物理パラメータ表示（欠陥情報など）などをもつ。CKD記録

- 23 -

オリエンテーション・テーブル28は、CKDに関連するプログラムがバッファ26を用いてデータを容易にアドレスし転送できるようにする仮想回転CKDバイト変位情報を示す。テーブル28は以下に示すいくつかの項目を含んでいる。

オリエンテーション・テーブル28

O R	方向設定ビット（論理または仮想方向設定）
C C	現シリンドラ・アドレス
C H	現ヘッド・アドレス
C S	現セクタ・アドレス
C F P	現フィールド・ポインタ
C F T	現フィールド形式（カウント、キーまたはデータ）
N C P	次のカウント・フィールド・ポインタ
P C P	前のカウント・フィールド・ポインタ
D L	バイト単位のCKDデータ・フィールド長
K L	バイト単位のキー・フィールド長
N M R	レコードなしビット

- 25 -

の制御部分にある関連するDASD物理パラメータが削除され、CKDバイト・アドレスを用いてレコードへのアクセスを可能にするのに十分正確な仮想トラックを論理的に定義するのに必要なパラメータだけが保持される。CKDカウント・フィールドは28バイトから12バイトに減少される。保持される12バイトは、CCHH（シリンドラ・ヘッド・アドレス）、R（レコード番号）、KL（バイト単位のキー長）及びDL（バイト単位のデータ長）を含む。さらに、仮想トラックの最後のレコードを識別するため、エミュレータ20がカウント・フィールドCFエミュレーションに1ビットを追加する。追加される1ビットLC65は、1に設定されると、カウント・フィールドが仮想CKDトラックの最後のカウント・フィールドであることを示す。仮想トラックにレコードが追加されると、1に設定されたLCビット65が最後に追加されたレコードに移る。他のすべてのLCビット65はゼロにリセットされる。LCビット65もROCF35に含まれる。

- 24 -

エミュレータ20はテーブル28を使って、実際のCKD DASD上のCKDデータへのアクセスに関連するシーク及び他のDASD物理活動を論理的にエミュレートする。第4図は、FBA DASD上で記録されエミュレートされるCKDデータにアクセスするデータ領域に関するエミュレーション活動を示す簡略化した流れ図である。DASDでの最初の動作は、アドレスされたトラックをシークすることである。したがって、チャネル実行部13から受け取ったSEEKチャネル・コマンドに応じて、シーク・エミュレート・ステップ70（テキストによる流れ図「SEEKコマンドの実行」を参照）が実行される。エミュレータ20は、SEEKコマンドと一緒に渡されたCC及びHHパラメータを反映するようにテーブル28を更新する。チャネル実行部13は、周知のように、SEEK CCWからSEEKコマンドを生成する。第2のステップ73は、回転オリエンテーションを獲得することである（上記の場合、回転（バイト変位識別）オリエンテーションがな

- 26 -

かったので OR = 0)。矢印 72 で示される SET SECTOR チャネル・コマンドにより、エミュレータ 20 は CKD 装置の SET SECTOR 動作をエミュレートする (テキストによる流れ図「SET SECTOR コマンドの実行」を参照)。このコマンドは、後続のデータ転送コマンドがアドレスされる回転位置 (バイト変位) を識別するだけである。オリエンテーション・テーブル 28 の OR ビットは依然として 0 であり、この時、データは DASD 16 からバッファ 26 に移動されていない。矢印 74 でチャネル実行部によって SEARCH ID チャネル・コマンドが発行されると、上位プロセッサ 10 は、データの転送を望んでいることを示す。このシーケンスのこの時点で、エミュレータ 20 が DASD 16 にアクセスして、ステップ 75 で示すように、DASD 16 からバッファ 26 にデータをステージングする。SEARCH ID の ID 部分が CKD アドレス・パラメータの識別を与えることに留意されたい。これらのパラメータは、どの FBA

- 27 -

ブロックをステージングし続けて、バッファにおいてディスクの回転をエミュレートする。通常、上位プロセッサは第 1 のアドレスされたレコードから始まる多数のレコードを読み取りまたは書き込む。この動作は、DASD 16 を用いた FBA 動作に関連する指標マークの有無に関わらず、仮想トラックの終端 (CKD データの終端) まで続く。チャネル・コマンドを受け取ると、それはデータ転送が停止するか、またはチャネル動作の終端に達したことを示す。上位プロセッサ・アクセスから独立したこの継続するステージングを矢印 79 で示す。

以下に、上記の動作の詳細を示すテキストによる流れ図を示す。流れ図の後の用語集で、使用した略語の意味を示す。これらの流れ図を実施するのに、どのプログラミング言語や論理ハードウェアを使ってもよい。実際の実施例では、例示したもの以外に、他のチャネル・コマンド実行コマンドを使ってマシンを完了してもよい。

- 29 -

ブロックが SEARCH ID コマンドに関連する CKD カウント・フィールド CF を記憶するかを識別するもの以外は、独立したアドレスには変換されない。この計算は簡単なので、数値例は示さない。

FBA ブロックがバッファ 26 に記憶されるとすぐ、エミュレータ 20 はステップ 78 でバッファ 26 にアクセスして、CKD DASD 上での CKD レコードの発見をエミュレートする。所期のレコードがバッファ 26 中で発見されると、オリエンテーションが行なわれたことを示す DE (装置終了) が上位プロセッサ 10 に送られる。オリエンテーション・テーブル 28 の OR ビットが 1 に設定される。上位プロセッサ 10 は、矢印 77 で示されるデータ転送コマンドを発行し、それにより、エミュレータ 20 がステップ 78 でバッファ 26 と主記憶装置 11 の間でデータを転送することにより DASD から読み取られた CKD をエミュレートする。こうしたデータ転送中に、エミュレータ 20 は DASD 16 からバッファ 26 に FBA

- 28 -

エミュレータ 2.0 のプログラム 25 を示す流れ図

以下に示すテキストによる流れ図は、本発明を例示するチャネル・プロセッサ 12 のマシン・ステップを示す。これらのステップは、完全なマシン設計を記述するものではない。完全なマシン中で見られる他のプログラムや構造は、本発明を実施するのに必要ではないからである。

呼び出されるサブルーチンを含むチャネル・コマンド実行の流れ図

SEEK コマンドの実行

受信コマンドの妥当性検査を行なう。

ファイル・マスクでシークが可能なことを確認する。

受信コマンドからシーク・パラメータを読み取り、様式を確認する。

シリンド (CC) アドレスとヘッド (HH) アドレスの妥当性検査を行なう。

シークが必要な (現 CCHH が要求された CCHH に一致しない) 場合は続行し、そうでない場合

- 30 -

は「チャネルに C E D E」ステップに進む。
C C H H をセーブし、それをシーク回路に入力する。
「シーク必要」ビット 1 0 0 を N S = 1 に設定する。
「オリエンテーション・ビット」を O R = 0 に設定する。
チャネルに C E D E する；連鎖を検査して、次に進む。

SET FILE MASK コマンドの実行

受信コマンドの妥当性検査を行なう。
受信コマンドからファイル・マスク・パラメータを読み取る。
ファイル・マスクをセーブする。
「オリエンテーション・ビット」を O R = 0 に設定する。
チャネルに C E D E する；連鎖を検査して、次に進む。

SET SECTOR コマンドの実行

受信コマンドの妥当性検査を行なう。

- 31 -

ションが行なわれ）かつ多トラック動作でない場合、ケース IV を実行する。
(最後のレコードを越えて、または最後のレコードの非カウント・フィールドにまでオリエンテーションが行なわれ) かつ多トラック動作が実行されている場合、ケース V を実行する。

5 つのケースの終わり

S I Z E = 受信パラメータのバイト数に設定する。
P T R から始めて S I Z E をパラメータと比較する。

比較の成否の結果をチャネルに送る；
比較が成功で、S I Z E < 5 の場合、S I Z E
5 としてセーブする。

状況の終了を行なう。

ケース I のステップ

P T R = C F P

C F P = C F P + 1 2

K L = 0 の場合は C F T = D F 、そうでない場合は C F T = K F

- 33 -

受信コマンドからパラメータを読み取る。
コマンドのセクタ番号の妥当性検査を行ない、それを S としてセーブする。
「オリエンテーション・ビット」を O R = 0 に設定する。

チャネルに C E D E する；連鎖を検査し、次に進む。

SEARCH ID コマンドの実行

受信コマンドの妥当性検査を行なう。
O R = 0 の場合、「GET ORIENTED」サブルーチンを呼び出す。
受信コマンドからパラメータを読み取る。
次の 5 つのケースの 1 つを実行する。
C F T = C N T の場合、ケース I を実行する。
C F T = R O C N T の場合、ケース II を実行する。
I D が最後のレコード以外のレコードの非カウント・フィールドを指す場合、または H A と R O を指す I D が存在する場合、ケース III を実行する。
(最後のレコードを越えて、または最後のレコードの非カウント・フィールドにまでオリエンテー

- 32 -

ケース II のステップ

P T R = C F P

C F P = C F P + 1 2

K L = 0 の場合は C F T = R O D F 、そうでない場合は C F T = R O K F

ケース III のステップ

P T R = N C P

K L = ポインタ P T R でのレコードのキー長

D L = ポインタ P T R でのレコードのデータ長

C F T = H A フィールドの場合は P C P = C F P 、そうでない場合は P C P = 次のカウント(次のカウント・ポインタを計算する)

C F P = P T R + 1 2

K L = 0 の場合は C F T = D F 、そうでない場合は C F T = K F

N C P = 次のカウント・ポインタを計算する(サブルーチンを呼び出す)

L C = 1 の場合は N M R = 真(1)、そうでない場合は N M R = 假(0)

- 34 -

ケースIVのステップ

EOT = 1 の場合は ERROR、そうでない場合は EOT = 1 (指標通過) に設定する

BB = (T * C + H) * B

BBLAST = (T * C + H) * B + (BPS * S / BBB) - 1 (用語集の BBB の項を参照)

S を > G1 / BPS かつ < (G1 + G2' + SIZE(HA) + PADH + ECCCH) / BPS に設定する。

ステップ4で GET ORIENTED を呼び出す。

GET ORIENTED = 0 の場合、レコードが見つからないことを示し、そうでない場合はレコードが見つかったことを示す。

PTR = CFP

CFP = CFP + 12

KL = 0 の場合は CFT = R0DF、そうでない場合は CFT = R0KF

ケースVのステップ

H = H + 1 に設定する。新しい H > T の場合、誤

- 35 -

ションが行なわれ、PO = 1 の場合、ケースAを実行する。そうでない場合は、

R0 に対してオリエンテーションが行なわれ、かつ(前のCMDが探索IDでも探索キーでもなく、R1が存在する)場合、ケースBを実行する。そうでない場合は、

HA に対してオリエンテーションが行なわれ、かつR0とR1が存在する場合、ケースCを実行する。そうでない場合は、

EOT = 1 で、多トラック動作でない場合、ケースDを実行する。そうでない場合は、

EOT = 1 で、多トラック動作の場合、ケースEを実行する。

ケースの終わり

位置 PTR からバイト数 L を読み取る。

L = 0 の場合、C E D E でチャネルに U C し、そうでない場合はチャネルに C E D E する。

ケースAのステップ

PTR = CFP

CFT = KF または R0KF の場合、PTR = P

- 37 -

りを知らせる。

S > (G1 / BPS) かつ < (G1 + G2' + SIZE(HA) + PADH + ECCCH) / BPS に設定する。

ステップ2から始めて「GET ORIENTED」を呼び出す。

レコードが見つからない場合、戻って S を設定する。そうでない場合は続行する。

PTR = CFP

CFP = CFP + 12

KL = 0 の場合は CFT = R0DF、そうでない場合は CFT = R0KF

読み取りデータ・コマンドの実行

受信コマンドの妥当性検査を行なう。

前のCMDが「SEARCH」で、比較して等しかった場合、PO = 1

OR = 0 の場合、GET ORIENTED を呼び出す。

非R0レコードに対してオリエンテーションが行なわれている(またはR0に対してオリエンテー

- 36 -

TR + KL。そうでない場合は、

CFT = CNT または R0CF の場合、PTR = PTR + 12 + KL

ステップYを実行する。

ステップY

L = DL

CFP = NCP

PCP = PCP から次のCNTを計算する。

NMR = 1 の場合、CFT = 指標(EOT)、そうでない場合はループを実行する。

ループ実行

CFT = CNT

KL = CFP の KL

DL = CFP の DL

NCP = CFP の CNT から計算する

LC = 1 の場合は NMR = 1、そうでない場合は NMR = 0

ケースBのステップ

CFP = NCP

L = CFP の DL

- 38 -

P T R = C F P + 1 2 + K L

P C P = C F P

C F P = C F P の C N T から次の C N T を計算する

ステップ X を実行する

ステップ X

P C P について L C = 1 の場合、 C F T = E O T 、
そうでない場合はループを実行する。

ループ実行

C F T = C N T

K L = C F P の K L

D L = C F P の D L

N C P = C F P から次の C N T を計算する

L C = 1 の場合は N M R = 1 、 そうでない場合は N M R = 0

実行を終了する。

ケース C のステップ

P T R = N C P から次の C N T を計算する

L = P T R の D L

C F P = P T R

- 39 -

ステップ Y を実行する。

ケース E のステップ

H = H + 1 と設定する。新しい H > T の場合、誤りを設定する。

S > (G 1 + G 2 ' + S I Z E (H A) + P A D H + E C C H) / B P S に設定する。

S < (G 1 + G 2 ' + S I Z E (H A) + P A D H + E C C H + G 2 + G 3 + S I Z E (C F) + P A D C + E C C C + 8 + P A D D + E C C D) / B P S に設定する。

G E T _ O R I E N T E D を呼び出す。ステップ 2 から始める。

N R F = 1 の場合 (レコードが見つからなかった場合) 、 ケース E の第 1 ステップに戻る。 そうでない場合は続行する。

P T R = C F P + K L + 1 2

ステップ Y を実行する。

ケース A ないし E の終わり

P T R から始めて L バイトをチャネルに送る。

L = 0 の場合、 U C をチャネルに C E D E する。

P T R = P T R + 1 2 + K L

P C P = C F P

ステップ X を実行する。

ケース D のステップ

E O T = 1 の場合、 E R R O R を設定する。 そうでない場合は E O T = 1 に設定する

B B = (T * C + H) * B

B B L A S T = (T * C + H) * B + (B P S * S / B B B) - 1 (B B B - 用語集を参照)

S > (G 1 + G 2 ' + S I Z E (H A) + P A D H + E C C H) / B P S に設定する

S < (G 1 + G 2 ' + S I Z E (H A) + P A D H + E C C H + G 2 + G 3 + S I Z E (C F) + P A D C + E C C C + 8 + E C C D) / B P S に設定する。

G E T _ O R I E N T E D を呼び出す。ステップ 4 から始める。

O R = 0 の場合、 レコードが見つからなかった。
そうでない場合はレコードが見つかった。

P T R = C F P + K L + 1 2

- 40 -

そうでない場合はチャネルに C E D E する。

W R I T E _ D A T A コマンドの実行

受信コマンドの妥当性検査を行なう。

C K D データを上位プロセッサから受け取り、主記憶装置 1 1 の個別バッファ領域に記憶する。以下のステップは、記憶された C K D データが第 3 図に示すように変換された後でそのデータに対しても行なわれる。

P T R = C F P

C F T = K F または R O K F の場合、 P T R = P T R + K L

ステップ Y を実行する。

P T R で上位プロセッサからの C K D データの L バイトをバッファ 2 6 に記憶する。

更新されたまたは新しい F B A ブロック (5 0 や 5 1 など) を D A S D 1 6 に書き込む。

すべての F B A ブロックを D A S D 1 6 に書き込むと、チャネルに C E D E を送る。別の動作に進む。

- 41 -

- 42 -

GET ORIENTED サブルーチン

このサブルーチンはオリエンテーション・テーブル28を管理する。

このコマンド連鎖中に SEEK コマンドがない場合、CC = CH = 0

このコマンド連鎖中に SET SECTOR コマンドがない場合、CS = 0

ステップ2

バッファ26にステージングされる、すなわち転送される最初のレコードを記憶する最初のFBA ブロックを識別する。

最初のFAB ブロックのアドレス = (T * C + H) * B + (BPS * CS / BBB) 下の整数に切り下げる。

DASD16からバッファ26にステージングされる最後のFBA ブロックを識別する。

最後のFBA ブロックのアドレス = (T * C + B) + (B - 1)

ステップ4

上記で識別されたFBA ブロック及びすべての中

- 43 -

BPS * S <= Z の場合、HA またはカウント・フィールドに対して方向設定される(セクタSを越えた場合は、ステップ9に飛ぶ。) そうでない場合は次に進む。

LC = 0 の場合、

Z' = Z + 12 + KL + DL + PADC + ECC
CC + PADD + ECCD + G2 + G3 に設定する。

KLがゼロでない場合は Z = Z' + PADK + ECCK + G2、そうでない場合は Z = Z'

(上記では、最初のレコードとEOTの間で探索が行なわれると仮定する)

ステップ9

NS = 0 (シークする必要はない) と設定する。
方向設定テーブル28を更新する。

CFP = Z

KL = レコードのカウント・フィールドのKL

DL = レコードのカウント・フィールドのDL

CC = レコードのカウント・フィールドのシリ
ンダ・アドレス

- 45 -

間のFBA ブロックを、第2図に示すようにバッファ26に読み込む。HDをFBA ブロック中のデータ(バイト2-2048)から分離しておく。
最初のFBA ブロックのHDを検査する;

FBA ブロック中にCFを指すポインタがない場合、最初のカウント・フィールドが見つかるまで後続のFBA ブロックHDを検査する。

カウント・フィールドが見つからずにEOTに達した場合、仮想トラックの始めから最初のFBA ブロックまでFBA ブロックをステージングして、カウント・フィールドを見つける。カウント・フィールドが見つからない場合、チャネルにUCを合図する。

FBA ブロックNR(最初のカウント・フィールドをもつブロック) = 最初のFBA ブロック - (T * C + H) * B と設定する。

Z = (NR * BBB) + HDからのオフセット・ポインタ(最初のカウント・フィールドの第1バイトがFBA ブロックNR内にあるバイト数)と設定する。

- 44 -

CH = レコードのカウント・フィールドのヘッド・アドレス

CFT = S * BPS < G1 ならば HA、そうでない場合は、G1 < S * BPS < G1 + G2 + SIZE(HA) + PADH + ECCH ならば R0 カウント・フィールド ROCF、そうでない場合はデータ・レコード「n」のCF。

OR = 1

LC = 1 の場合、NMR = 1 と設定する。そうでない場合は NMR = 0

呼出しプログラムに戻る。

次のカウント・フィールド・ポインタの計算の
サブルーチン

ポインタPTR2はエミュレートされた仮想トラック中の次のカウント・フィールド位置を示す。

PTR = G1 の場合、PTR2 = PTR + 12 + PADH + ECCH + G2'、そうでない場合は PTR2 = PTR + 12 + (PTR が指すレコードの) KL + (PTR が指すレコードの) DL + PADC + ECCC + PADD + ECC

- 46 -

D + G 2 + G 3

(PTRが指すレコードの) KL ≠ 0 の場合、
PTR 2 = PTR 2 + G 2 + PADK + ECC
K

PTR 2 を呼出しプログラムに戻す。

エミュレーションのために FBA ディスクを初期化するプログラム

CKD ディスクの FBA ブロックの数

X = CKD トランクのバイト数と置く。

B = X / BBB (各 CKD トランク中で使用されるブロックの数) と置く。

CKD 装置 1 台当たりのトラック数は、シリンドの数とシリンド 1 個当たりのトラック数の積である。各 CKD 装置のブロックの数は、CKD 装置のトラック数と B の積である。

CKD エミュレーションのために FBA ディスクを初期化するステップ

CYL = エミュレートされる CKD 装置のシリンド数と置く。

まずエミュレーションに使用される FBA ディス

- 47 -

= 8 及び LC がすべて 1 である点以外は上記の HA の場合と同様に R0 を作成する。

FBA ディスクの FBA ディスク上で B - 1 個のブロックで分離された適切なブロックに現バッファからのブロックを書き込む。

CKD トランクを B 個の FBA ブロックに書き込むステップ

この流れ図は、CKD トランクから FBA ディスク上の B 個の FBA ブロックにデータをコピーすることに関連している。

バッファ項目 (番号 0 ないし B - 1) でバッファ 26 を構成する。

初期化された主記憶装置の B フラグ・ビット BF 101 を 0 (偽) に設定する。所定の FBA ブロック (バッファ 26 項目) について、そのフラグ BF = 1 の場合、そのフラグ・ビットに関連する FBA ブロックに制御情報が記録されている。バッファ 26 中で、当該のすべての HD の第 2 バイトに 2 進パターン 00001000 を記憶する。このパターンは、データが当該の FBA ブロックに

- 49 -

クのすべての FBA ブロックを初期化して、使用される各 FBA ブロックのバイト 0 と 1 に 16 進数 0008 を記憶させる。さらに、B 番目ごとのブロックが論理 HA と R0 レコードを含む。この含有について以下に記述する。

HA と R0 で初期化される FBA ブロックのすべてについて実行する。

バッファ 26 を、繰り返して使用される FBA ブロックに設定する。

バイト 0 と 1 に、最初の 12 ビットの G1 と最後の 4 ビットのすべての 0 を記憶する。

各ブロックの G1 + 2 のバイト・オフセットの所に、HA を次のように書き込む; フラグ・バイトを 0, CC = 0 から CYL - 1 まで増分されるシリンド番号、HH = ゼロから T - 1 まで増分されるトラック番号、R バイト (レコード番号) = 0, KL = 0, DL = 0, LC = 0, パッド・バイト = 0

G1 + G1' + 2 (HA) + ECCH + PADH の各ブロックのバイト・オフセットの所に、DL

- 48 -

記録されていないことを示す。

CB102 は、バッファ 26 項目に書き込まれる次のバイトを指す。最小値は 2 である (HB バイト 0 と 1 を飛び越す)。

CKD トランクは、フィールド HA, ROCF, RODF, CF1, KF1, DF1 などをもつ主記憶装置中の CKD トランクとしてコピーされている。

メモリ・ポインタはすべてゼロである。

カウント・フィールドは 28 バイトではなく 12 バイトしかない。CKD 物理パラメータなどが削除されている。

書き込みが呼び出される度に、CB から始めてバイトが書き込まれ、HD バイトは常に飛び越される。カウント・フィールドが書き込まれ、それに対応するフラグ BF がオフの場合、そのフラグ BF はオンになる。対応する HD は、書き込まれたカウント・フィールドに対するオフセットを含むようにされる。CB は常に更新される。

- 50 -

CKDをバッファ項目に書き込むステップ

C B = 2

G 1 ゼロ・バイトを書き込む（各バイトが記録されるとき C B を更新する）。

H A を書き込む。

G 2' + S I Z E (H A) - 1 2 + P A D H + ゼロ・バイトの E C C H を書き込む。

R O C F を書き込む。

R O D F を書き込む。

G 2 + G 3 + S I Z E (C N T) - 1 2 + P A D C + E C C C + P A D D + ゼロ・バイトの E C C D を書き込む。

レコード R 0 以外の各 CKD レコードについて繰り返す。

C F を書き込む。

K F を書き込む。

D F を書き込む。

G 1 + G 2 + G 3 + S I Z E (C N T) - 1 2 + P A D C + E C C C + P A D K + E C C K + P A D D + ゼロ・バイトの E C C D を書き込む。

- 51 -

C E チャネル終了信号、チャネルが開放されることを示す

C E D E チャネル終了、装置終了、チャネル動作の完了を示す

C F CKD 様式レコードのカウント・フィールド

C F P 現フィールド・ポインタ

C F T 現フィールド形式

C I D F B A 装置中のヘッダのバイトの数

C K D カウント、キー、データ

C M D チャネル・コマンド

C Y L 1 つの CKD D A S D 中のシリンド数

D E 装置終了、装置が動作を完了したことをチャネルに合図する

D F CKD 様式レコードのデータ・フィールド

D L データ・フィールド D F の長さ、カウント・フィールド C F 中のフィールド

E C C エラー検出・訂正冗長バイト

E C C C カウント・フィールド C F 中の E C C バイトの数

- 53 -

D A S D 1 6 の B 個の F B A ブロックにバッファ 2 8 を書き込む。

用語集

B 1 つの C K D 仮想トラックに含まれる F B A ブロックの数

B B 現バッファにステージングされる最初の F B A ブロックのブロック番号。 B B B : 値 B P B - C I D

B B L A S T C K D ステージング動作で現バッファにステージングされる最後のブロックの F B A ブロック番号

B F フォーマットされたかどうかを示すブロック b に関連するフラグ

B P B F B A ブロック 1 個当たりのバイト数

B P S C K D データ様式のセクタ 1 個当たりのバイトの数

C B 処理中の現バイトを指すポインタ

C C エミュレートされる現バイトのシリンド番号またはアドレス

C C W チャネル・コマンド（制御）ワード

- 52 -

E C C D データ・フィールド D F 中の E C C バイトの数

E C C H H A 中の E C C バイトの数

E C C K キー・フィールド K F 中の E C C バイトの数

E O T トラック終端、物理 D A S D の指標に関連する

F B A 固定ブロック・アーキテクチャ

G X バイト中のギャップ長、ただし x = 1 ないし 3 、 C K D 様式

H H D A S D (C K D または F B A) のヘッドまたは表面アドレス

H A C K D 様式のホーム・アドレス・レコード

H D C K D エミュレーション用の F B A ブロックのヘッダ（制御フィールド）

K F C K D 様式のキー・フィールド

K L キー長、K L を示すカウント・フィールド中 C F の標識

N C P 次のカウント・フィールド・ポインタ

N R アクセスされる次の C K D レコード

- 54 -

N R F 探索基準に合致するレコードが見つからない
N S ビットまたはフラグをシークする必要がある
P A D C カウント・フィールド C F 中の埋込みバイトの数
P A D D データ・フィールド D F 中の埋込みバイトの数
P A D H H A 中の埋込みバイトの数
P A D K キー・フィールド K F 中の埋込みバイトの数
P C P 前のカウント・フィールド・ポインタ
P D S 区分データ
P O 以前に論理的にオリエンテーションが行なわれた
P T R 処理中のフィールド／項目を指すのに使われるポインタ
R レコード番号、フィールドはカウント・フィールド C F 中にある
S セクタ番号

- 55 -

S I Z E エミュレートされる C K D 装置のバイトで表わしたフィールド（次の項目で識別されるフィールド）のサイズ

T D A S D の 1 つのシリンドラのトラック数

U C 単位検査、誤りを示すチャネルへの信号

V T O C D A S D の内容のボリューム・テープル

E. 発明の効果

本発明は、エミュレートされるレコードのアクセス性能を改善する。

4. 図面の簡単な説明

第1図は、本発明が有利に実施できる情報処理システムの簡略化した図である。

第2図は、第1図に示した情報処理システムで本発明を実施する構成図である。

第3図は、第1と第2のデータ様式及び第2図に示すように実施された第2のデータ様式で第1のデータ様式のエミュレーションの構成図である。

第4図は、第2図で使用されるマシン動作の簡略化した流れ図である。

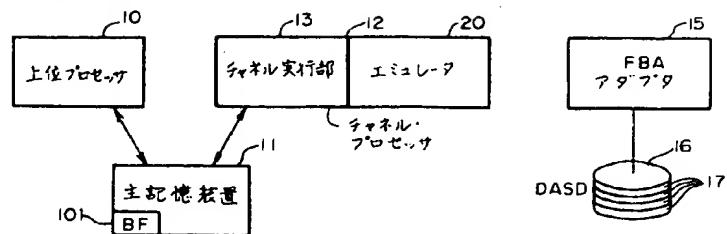
- 56 -

1 0 ……上位プロセッサ、 1 1 ……主記憶装置、
1 2 ……チャネル・プロセッサ、 1 3 ……チャネル実行部、 1 5 ……F B A アダプタ、 1 6 ……D A S D、 2 0 ……エミュレータ、 2 5 ……エミュレータ・プログラム、 2 8 ……仮想トラック・バッファ、 2 8 ……オリエンテーション・テーブル。

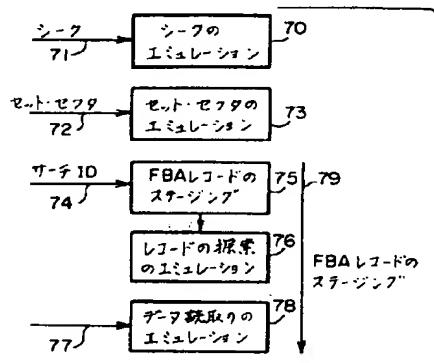
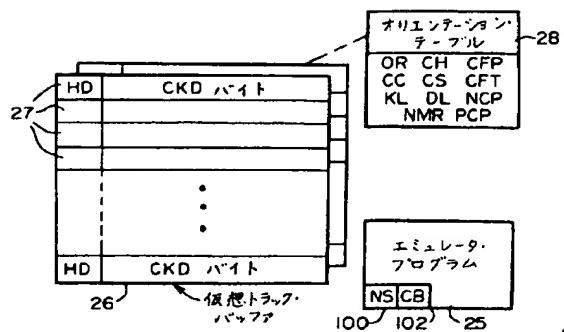
出願人 インターナショナル・ビジネス・マシーンズ・コーポレーション
代理人 弁理士 山 本 仁 朗
(外 1 名)

- 57 -

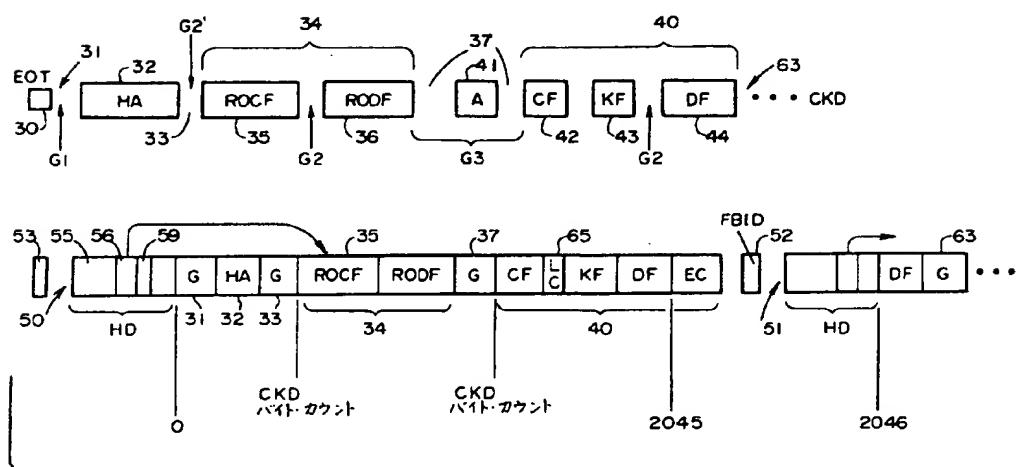
第 1 図



第 2 図



第 4 図



第 3 図